Real-Time Hardware-in-the-Loop Testing with Common Simulation Framework

Judith D. Gardiner
Ohio Supercomputer Center, Columbus, OH
614-292-9623
judithg@osc.edu

Abstract

The Common Simulation Framework (CSF) facilitates live, virtual, and constructive (LVC) simulation of a complete missile deployment system. It includes some support for real-time operation but not enough to allow hardware-in-the-loop (HWIL) testing. This project added real-time monitoring capabilities to CSF and demonstrated its use for a HWIL simulation. The demonstration involved driving a motion simulator table at the Redstone Technical Test Center in Huntsville, AL. The project successfully showed the effectiveness of CSF in a real-time HWIL environment.

Introduction

This project extends the real-time capabilities of the Common Simulation Framework (CSF) and demonstrates the effectiveness of CSF for hardware-in-the-loop (HWIL) testing. Our objectives were twofold, to enhance the real-time support provided in CSF and to provide example code for a HWIL simulation using CSF. As a simple demonstration we used CSF to drive a motion simulator table through a preprogrammed trajectory.

Past work in this area involved changes to CSF that were unique to the local operating system and hardware under test. Our goal was to work within the CSF framework, creating software that can be efficiently reused and adapted for other simulations. We made minimal changes to CSF, instead creating plug-in modules within the CSF framework. The extensions and modifications to CSF will be submitted for possible inclusion in the standard distribution.

This work was done as part of the PET program under the High Performance Computing Modernization Program.

Hardware and Software Environment

Testing on this project was done at the AMSTAR Production Facility at Redstone Technical Test Center (RTTC) in Huntsville, AL. The test environment included a motion simulator table, a table control computer, and a computer running the simulation. The computers communicate by way of a reflective memory network.

The computer running CSF was an Intel® Core™2 Quad processor, a four-core multiprocessor. It runs the RedHawk operating system, which is a real-time version of Linux. RedHawk is based on a multithreaded, fully preemptible Linux kernel with low-latency enhancements.

The simulation sends trajectory information in real-time to the table controller via reflective memory. Reflective memory is a real-time networking technology designed to mimic shared memory. It has very predictable latencies and transfer times. Access to reflective memory is handled through a locally developed program called RAP (Resource Allocation Program).

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **JAN 2009**	2. REPORT TYPE	3. DATES COVERED **00-00-2009 to 00-00-2009**

4. TITLE AND SUBTITLE **Real-Time Hardware-in-the-Loop Testing with Common Simulation Framework**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Ohio Supercomputer Center, 1224 Kinnear Road,Columbus,OH, 43212**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Live-Virtual Constructive Conference, 12-15 Jan 2009, El Paso, TX

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**3**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Common Simulation Framework
CSF is a simulation framework developed by AMCOM RDEC (Army, Huntsville) to create a unified way to develop and integrate models for missile flight simulations. It is intended to be general enough to be used in a variety of computational environments and to support a wide range of simulation domains. CSF provides some support for real-time operation, but not enough to allow hardware-in-the-loop testing.

CSF is an object-oriented simulation development environment. It supports the visual assembly of component models into composite models of arbitrary complexity. CSF allows users to develop component models as software "plug-ins" and assemble them into a complete simulation. We used the current distribution of CSF, CSF 5.2, released Aug. 8, 2007.

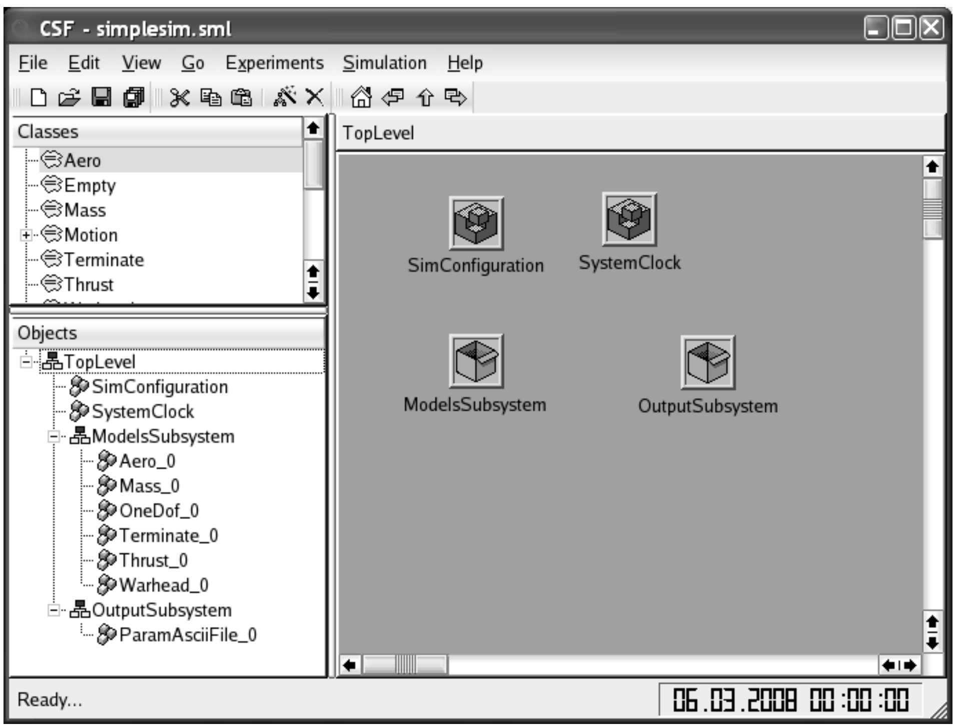

Figure 1: CSF GUI showing a simple missile simulation with two subsystems.

Real-Time Extensions to CSF
When running a simulation with hardware in the loop, or in any other hard real-time environment, it is critical to ensure that all real-time deadlines are met. Updates must be sent to the hardware at regular intervals. If some portion of the simulation takes too long and an update is not ready at the specified time, the simulation results will not be valid. There may even be a risk of hardware damage if the simulation is allowed to continue. It is therefore essential to monitor for frame overruns and provide the ability to terminate the simulation automatically and gracefully if an overrun occurs.

CSF does not have a real-time monitoring capability built in, but it does provide the hooks for adding such a capability as a plug-in library. We have developed a real time monitor class called

RealTimeMonitor that can be run as part of any real-time simulation. It can be run in a separate thread to eliminate any performance penalty. The monitor reports on frame overruns and can be configured to terminate the simulation if an overrun is detected.

Hardware-in-the-Loop Demonstration

We demonstrated the effectiveness of CSF for real-time HWIL simulation by using it to drive a motion simulator table. The demonstration used a preprogrammed six-degree-of-freedom (6DOF) trajectory with an update rate of 600 Hz. The trajectory points were read from a file and sent to the motion table controller at the appropriate real-time rate. The real-time monitor was run with the simulation. Duty cycles were less than 2% for this very simple simulation.

Several considerations should be kept in mind when running a real-time simulation under CSF. It is necessary to replace the default system clock with a suitable real-time clock, and the clock rate must be set at a value that can be sustained by the system in real time. Preliminary testing can be done using the GUI, with slow clock rates, but actual real-time operation must be done in batch mode, which has much less overhead. Finally, it may be beneficial to select Euler integration, which performs less computation per time step than the other integration strategies.

Conclusion

The goals of this project involve both real-time performance and generality of the software solution. We successfully demonstrated that CSF can be used in a real-time HWIL environment. Furthermore, our software solution is general enough to be of use to others in the DoD test and simulation community. The real-time monitor software we developed can be used directly in other simulations, while the demonstration software serves as a starting point for code development by other users.

The extensions and modifications to CSF will be offered to the CSF steering committee for acceptance in the standard distribution. If accepted, the software will be available to a wide range of users across DoD's test and range facilities.

Acknowledgements

This publication was made possible through support provided by DoD HPCMP PET activities through Mississippi State University under contract GS04T01BFC0060. The opinions expressed herein are those of the author(s) and do not necessarily reflect the views of the DoD or Mississippi State University.